

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Programa de Pós-Graduação em Educação
Evandro José de Oliveira Júnior

PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE APRENDIZAGEM PARA O
ENSINO DE LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA

Diamantina
2017

Evandro José de Oliveira Júnior

**PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE APRENDIZAGEM PARA O
ENSINO DE LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao programa de Pós-Graduação em Educação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito para obtenção do título de Mestre.

Orientador: Alexandre Ramos Fonseca

**Diamantina
2017**

Ficha Catalográfica - Sistema de Bibliotecas/UFVJM
Bibliotecário Anderson César de Oliveira Silva, CRB6 – 2618.

O48p Oliveira Júnior, Evandro José de.
2017 Plataforma de Suporte às Ferramentas de Aprendizagem para o
 Ensino de Lógica de Programação na Educação Básica / Evandro
 José de Oliveira Júnior – Diamantina, 2017.
 57 p. : il.

Orientador: Prof. Dr. Alexandre Ramos Fonseca

Dissertação (Mestrado Profissional - Programa de Pós-
Graduação em Educação) –Universidade Federal dos Vales do
Jequitinhonha e Mucuri.

1. Educação Básica. 2. Ciência da computação. 3. Lógica de
Programação. 4. Interdisciplinaridade. 5. Blockly. I. Fonseca,
Alexandre Ramos. II. Título.

CDD 371.334

Elaborada com os dados fornecidos pelo(a) autor(a).

Evandro José de Oliveira Júnior

**PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE APRENDIZAGEM PARA O
ENSINO DE LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA**

Dissertação apresentada ao programa de Pós-Graduação em Educação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisito para obtenção do título de Mestre.

Orientador: Prof. Dr. Alexandre Ramos Fonseca

Data de aprovação ____/____/____.

Prof. Dr. Alexandre Ramos Fonseca
Instituto de Ciência e Tecnologia - UFVJM

Prof. Dr. Euler Guimaraes Horta
Instituto de Ciência e Tecnologia - UFVJM

Prof. Dr. Paulo Cesar de Resende Andrade
Instituto de Ciência e Tecnologia - UFVJM

Diamantina
2017

AGRADECIMENTOS

Agradeço a Deus por ter guiado meus passos dando sabedoria, persistência e força. Aos meus pais, Evandro e Doralice, aos meus irmãos Leonardo, Eduardo, Maria Tereza e João Pedro, pessoas essenciais em minha vida. A minha amada noiva Grace pela paciência e apoio. Aos meus amigos e aos colegas do PPGEd que sempre estiveram dispostos a ajudar. Aos Professores Alexandre e Euler que me passaram todo conhecimento necessário para seguir com este trabalho. À UFVJM pela oportunidade de estudar em cursos qualidade. Ao PPGEd e todos os docentes por terem me aceitado como aluno. Ao SENAC Montes Claros, IFNMG / Pirapora e a Fundação Educacional Montes Claros sem essas instituições não seria possível concretizar este sonho.

RESUMO

Alguns países são referência no ensino de computação em escolas de ensino básico. Isso se deu pelo fato de não ensinarem somente ferramentas básicas contidas nos computadores, mas também os fundamentos da Ciência da Computação e como eles podem ser aplicados no dia-a-dia. No Brasil praticamente não há ensino de computação nas escolas. Com o objetivo de estimular a inserção da computação na educação básica do Brasil, este trabalho apresenta várias ferramentas de ensino e aprendizagem de programação. Discute como essas ferramentas podem ajudar os alunos a desenvolverem características e conhecimentos desejáveis, tanto para a própria computação quanto na sua interdisciplinaridade. É realizada uma análise sobre as características de duas dessas ferramentas: *Blockly* e *App Inventor*. Além disso, são apresentados exemplos de uso. Mostra também uma reflexão sobre a importância de trabalhar e desenvolver o pensamento crítico e o raciocínio lógico dos alunos através da lógica de programação inserida nas escolas. Foi proposto nesta dissertação o desenvolvimento de uma plataforma que reúna as principais ferramentas e oriente os alunos e professores sobre como usá-las. Este trabalho busca contribuir para escola, alunos e professores, apresentando ferramentas de ensino que interajam com crianças e adolescentes, de forma a proporcionar diversão e estimular a curiosidade.

Palavras-chave: Educação Básica. Ciência da computação. Lógica de Programação. *Blockly*. *App Inventor*. Interdisciplinaridade.

ABSTRACT

Some countries are a reference in computer science education in elementary schools. This is due by the fact they are not teaching only basic computers tools, but also computer science fundamentals and how they can be applied in their daily life. In Brazil, there is practically no computer science education in schools. With the aim of motivate the insertion of computing science in Brazil's basic education, this work presents several teaching and learning tools for computer programming. It discusses how these tools can help students to develop characteristics and knowledge desirable, for their own programing skills and also in their interdisciplinary. An analysis is performed on the characteristics of two tools: Blockly and App Inventor. In addition, their usage examples are provided. It also shows a reflection on the importance of developing the student critical and logical thinking using programming logics inserted in schools. It was proposed in this dissertation the development of a platform that brings together the main tools and guides students and teachers on how to use them. This work aims to contribute to school, students and teachers, presenting teaching tools which interacts with children and teenagers, in order to provide fun and stimulate curiosity.

Keywords: Basic Education. Computer Science. Programming Logic. Blockly. App Inventor. Interdisciplinarity.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Exemplo de algoritmo para preparar um bolo usando lista de comandos . . . | 27 |
| Figura 2 – Exemplo de algoritmo para preparar um bolo usando lista de comandos . . . | 28 |
| Figura 3 – Exemplo de algoritmo para preparar um bolo usando lista de comandos . . . | 29 |
| Figura 4 – Exemplo de algoritmo para preparar um bolo usando programação em blocos | 30 |
| Figura 5 – Scratch, ferramenta de ensino de lógica de programação para crianças . . . | 31 |
| Figura 6 – Tela inicial do <i>Blocly Games</i> com os sete jogos disponíveis | 33 |
| Figura 7 – <i>Blocly Games</i> : Jogo do labirinto | 34 |
| Figura 8 – Uso de bloco de repetição | 35 |
| Figura 9 – Uso de bloco condicional | 35 |
| Figura 10 – Algoritmo que solucionaria qualquer labirinto | 36 |
| Figura 11 – Código em linguagem JavaScript correspondente ao algoritmo da Figura 10 | 37 |
| Figura 12 – <i>App Inventor</i> : Modo de design | 38 |
| Figura 13 – Interface do aplicativo de cores executado em um celular com sistema Android | 39 |
| Figura 14 – <i>App Inventor</i> : Modo de desenvolvimento de lógica de programação utilizando blocos | 39 |
| Figura 15 – Plataforma Apcode: Tela inicial | 46 |
| Figura 16 – Plataforma Apcode: Tela onde são listadas as ferramentas de ensino | 47 |
| Figura 17 – Plataforma Apcode: Tela de suporte ao aluno | 48 |
| Figura 18 – Plataforma Apcode: Tela da área reservada aos professores | 49 |
| Figura 19 – Plataforma Apcode: Tela da página sobre | 50 |
| Figura 20 – Blog da plataforma Apcode | 51 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Ferramentas de ensino de lógica de programação que utilizam metodologia de blocos | 32 |
|--|----|

LISTA DE ABREVIATURAS E SIGLAS

UFVJM - Universidade Federal dos Vales do Jequitinhonha e Mucuri

IoT - *Internet of Things*

PISA - *Programme for International Student Assessment*

UFMG - Universidade Federal de Minas Gerais

MIT - *Massachusetts Institute of Technology*

BBC - *British Broadcasting Corporation*

SBC - Sociedade Brasileira de Computação

LDB - Lei de diretrizes e bases da educação nacional

PDE - Programa de Desenvolvimento Educacional

URL - *Uniform Resource Locator*

LISTA DE SÍMBOLOS

® Marca registrada

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 21 |
| 1.1 | Objetivos | 23 |
| 1.1.1 | Objetivo Geral | 23 |
| 1.1.2 | Objetivos Específicos | 23 |
| 1.2 | Organização | 23 |
| 2 | ENSINO DE CIÊNCIA DA COMPUTAÇÃO | 25 |
| 3 | ALGORITMOS E SUAS FORMAS DE REPRESENTAÇÃO | 27 |
| 4 | FERRAMENTAS DE ENSINO DE LÓGICA E PROGRAMAÇÃO UTILIZANDO BLOCOS | 31 |
| 4.1 | <i>Blockly Games</i> | 33 |
| 4.2 | <i>Ambiente App Inventor</i> | 37 |
| 5 | LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA | 41 |
| 6 | INTERDISCIPLINARIDADE DA LÓGICA DE PROGRAMAÇÃO . . | 43 |
| 7 | PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE ENSINO DE LÓGICA DE PROGRAMAÇÃO | 45 |
| 8 | CONSIDERAÇÕES FINAIS E TRABALHO FUTURO | 53 |
| | REFERÊNCIAS | 55 |

1 INTRODUÇÃO

O mundo atual passa por uma verdadeira revolução digital. A grande maioria das pessoas têm contato com variados tipos de tecnologia em diversos dispositivos que visam a facilitar o dia a dia. As crianças de hoje têm acesso a *tablets*, celulares, computadores, videogames, etc. Isso tudo só foi possível em virtude do avanço exponencial da Computação nas últimas décadas. Carros, geladeiras, TVs entre outras “coisas” estão sendo conectadas através da rede. Segundo Rose, Eldridge e Chapin (2015), essa área é conhecida como internet das coisas (*Internet of Things* - IoT). Em um futuro próximo será inevitável a necessidade de um conhecimento básico de computação para que as pessoas interajam de forma plena com essas novas tecnologias.

O ensino de computação no Brasil é direcionado apenas para cursos técnicos e de graduação. Por outro lado, a informática vem sendo adotada em muitas escolas, seja como disciplina obrigatória ou complementar. O ensino da informática abrange apenas o uso das tecnologias sem se aprofundar em como elas são concebidas ou como funcionam. Ensina-se basicamente o uso de algum sistema operacional, ferramentas de escritório e o uso da Internet. Em contrapartida, a Ciência da Computação envolve lógica, programação, desenvolvimento de sistemas, bancos de dados, funcionamento do hardware, entre outros conhecimentos. No Brasil, segundo Scaico *et al.* (2012), o ensino de computação nas escolas fundamentais é algo raro, o que pode ser um dos motivos da falta de interesse na área. Moran (2001) revela que, quando há, o ensino tecnológico nas escolas é falho com os alunos, uma vez que eles devem ser educados não só na aplicação e uso de tecnologias digitais (Informática), mas na forma como ela funciona e é concebida (Ciência da Computação). Papert, Valente e Bitelman (1980) afirmam que ao invés do computador programar a criança, a criança que deve programar o computador. Assim, torna-se muito importante o ensino de tal ciência para alunos de escolas do ensino fundamental e médio.

O programa Internacional de Avaliação de Estudantes de 2015 (PISA, na sigla em inglês), divulgou recentemente que o Brasil caiu no ranking mundial de educação nas três áreas avaliadas: ciências, leitura e matemática. O ensino de computação possibilita trabalhar características como o raciocínio lógico, tomada de decisão e pensamento crítico dos alunos, nas diversas áreas que abrangem a base curricular de ensino do Brasil. Isso demonstra potencial para melhorar o índice brasileiro no PISA, além de seguir uma tendência mundial de preparar os jovens para essa ciência (SCHLEICHER, 2015).

Ensinar computação é extremamente importante para o desenvolvimento do pensamento lógico. A programação, uma das subáreas da Ciência da Computação, consiste em escrever um conjunto de regras de acordo com alguma linguagem que resultará em um programa/aplicativo capaz de solucionar um problema. Dessa forma, programar é usar o fluxo de informações do problema para chegar em uma determinada solução, o que pode ser aplicado em várias áreas de conhecimento.

Para a academia científica independente do Reino Unido (The Royal Society), em

um relatório de 2012 intitulado “*Shut down or restart?*” (“Desligar ou reiniciar?”, em tradução livre) sugeriu que uma boa compreensão de Ciência da Computação estimula o raciocínio lógico e tomada de decisão. De acordo com Furber (2012) na Inglaterra, referência no ensino de computação para crianças, a Ciência da Computação é uma disciplina obrigatória, ela faz parte do currículo do ensino regular. Outros países também implantaram a computação na grade curricular das escolas fundamentais. Nos Estados Unidos (TUCKER, 2003) e Nova Zelândia (BELL; ANDREAE; ROBINS, 2012), os governos reconheceram a importância do impacto na economia de seus países, visando a facilidade que as crianças terão em adaptar mais facilmente às mudanças. Em alguns países que já inseriram a computação nas escolas básicas, o ensino começa a partir dos 6 anos de idade (OLSON, 2012).

Em algumas escolas do Rio Grande do Sul a computação também é apresentada como disciplina. Os alunos cursam introdução a computação uma vez por semana, aprendendo lógica, “atividades desplugadas”¹, desenvolvimento de jogos, histórias e animações. Outras iniciativas também no Brasil podem ser citadas. São escolas voltadas para o ensino de programação, robótica e áreas afins. A Loopye (PIZARRO, 2016), startup criada por alunos da UFMG, mantém um projeto voluntário de ensino de computação em uma escola pública de Belo Horizonte. Os sócios desenvolveram uma ferramenta para o ensino de lógica de programação. A Happycode (KLEINA, 2016), surgida nos Estados Unidos, já conta com algumas franquias no Brasil. Ela utiliza algumas ferramentas de ensino disponíveis na *internet*. Outra escola, a Buddys (ISIS, 2016), também tem obtido sucesso no ensino de programação e robótica para crianças.

A Ciência da Computação evolui rapidamente e sua interdisciplinaridade ocorre de forma natural. Um desafio significativo é como preparar os professores para o ensino dessa ciência. Para tanto, existem ferramentas onde os alunos podem interagir com os conceitos de lógica e programação auxiliadas pelos docentes. Essas ferramentas possuem interfaces lúdicas, o que pode despertar maior interesse pelo aprendizado.

Grandes empresas apoiam o ensino de Ciência da Computação no ensino básico. Organizações como Google®², Microsoft®³, Facebook®⁴ e MIT⁵ (Instituto de Tecnologia de Massachusetts), são responsáveis por investimentos e participam diretamente no desenvolvimento de ferramentas de ensino de programação e Ciência da Computação.

Para aprender a programar é requerido conhecimentos avançados como: domínio de uma linguagem de programação, conceitos matemáticos, construção de códigos, ambientes de programação entre outros. Para tanto, deve-se ter o cuidado necessário para ensinar programação de forma gradativa para que se desperte o interesse nos alunos. Nesse contexto se enquadram ferramentas lúdicas de ensino. Na sua grande maioria, essas ferramentas utilizam uma metodologia parecida com um jogo de quebra-cabeça, que consiste em encaixar blocos (comandos básicos de

¹ São atividades de ensino que não utilizam o computador ou qualquer outro tipo de tecnologia.

² <https://developers.google.com/blockly/>

³ <http://microbit.org/>

⁴ <https://code.org/>

⁵ <http://appinventor.mit.edu/explore/>

programação) para determinar uma solução.

A programação em blocos utiliza elementos gráficos interativos que se assemelham a peças de encaixe. As peças representam comandos contidos nas linguagens de programação. O objetivo é arrastá-los para uma área de trabalho e encaixá-los de forma a produzir uma sequência lógica, que é a solução do problema proposto pela ferramenta. Outra característica determinante dos softwares de ensino de programação é que eles atuam com níveis graduais de dificuldade, cada um fornecendo uma base para o próximo. Assim, a cada nível desenvolvido, a complexidade do problema aumenta e novos conceitos são apresentados. Este trabalho se baseia em experiências positivas obtidas em outros países, principalmente na Inglaterra, que já relata bons resultados ao ensinar os alunos a resolver problemas utilizando técnicas computacionais.

1.1 Objetivos

A seguir são apresentados os objetivos desta dissertação.

1.1.1 Objetivo Geral

Este trabalho tem como objetivo apresentar ferramentas de ensino e aprendizagem de programação, suas características específicas e como as ferramentas podem ser utilizadas em outras áreas de conhecimento. É proposto também o desenvolvimento de uma plataforma *on-line* que reúne as principais ferramentas de ensino de lógica de programação que utilizam blocos de encaixe. A plataforma também visa a fornecer suporte aos alunos e professores.

1.1.2 Objetivos Específicos

Como objetivos específicos para esta dissertação têm-se:

1. Discutir a importância de ensinar lógica de programação para crianças e em especial para alunos de ensino básico das escolas do Brasil;
2. Analisar as formas de apresentação de um algoritmo e encontrar a melhor forma para ensinar crianças e adolescentes;
3. Apresentar ferramentas de ensino que utilizam metodologia de blocos de encaixe;
4. Analisar como algumas dessas ferramentas funcionam;
5. Defender o ensino de lógica de programação nas escolas básicas do Brasil e a utilização de forma interdisciplinar.

1.2 Organização

Este trabalho está organizado da seguinte maneira: na Seção 2 é apresentado um panorama sobre o ensino de Ciência da Computação e sua importância no desenvolvimento do

raciocínio lógico dos alunos; na Seção 3 são mostradas as formas que um algoritmo pode ser representado e o que é programação em blocos; na Seção 4 são listadas várias ferramentas de ensino de lógica de programação que utilizam a metodologia de blocos; na Seção 5 discute-se o ensino de lógica de programação na educação básica; na Seção 6 a utilização das ferramentas de ensino em outras disciplinas escolares; Na Seção 7 é proposta uma plataforma de suporte às ferramentas voltado para alunos e professores; e na Seção 8 são apresentadas as considerações finais e trabalhos futuros.

2 ENSINO DE CIÊNCIA DA COMPUTAÇÃO

Ciência da Computação, Engenharias e outros cursos de ciências exatas e tecnológicas têm cada vez mais a computação como ferramenta básica. O perfil do aluno brasileiro que ingressa nessas carreiras é a de um estudante que possui afinidade com física, química, matemática e possui noções de informática. Normalmente, esses alunos não possuem um embasamento mínimo sobre os princípios da computação. Algumas disciplinas de computação e introdução à programação (algoritmos) são ministradas nos primeiros períodos de cursos de graduação da área de Ciências Exatas. Muitas vezes essas disciplinas não são atrativas para os alunos por serem apresentadas de maneira muito técnica. Provavelmente, esse será o primeiro contato do aluno com os princípios da computação. Isso pode ser frustrante, uma vez que o aluno até então teve apenas contato com as ferramentas básicas da informática.

Ensinar programação não é somente ensinar uma linguagem de programação. Santos e Costa (2006) revelam que antes de ensinar uma linguagem de programação é preciso desenvolver o raciocínio lógico, que será necessário para a resolução de problemas computacionais. Em virtude disso, o trabalho de desenvolver o raciocínio lógico deveria iniciar na educação básica e de maneira adequada, como tem sido feito em alguns países, de maneira lúdica e gradativa.

Hoje, grande parte das crianças e jovens têm familiaridade com dispositivos tecnológicos como computadores, celulares e *tablets*. Dessa forma, a computação torna-se algo rotineiro no dia a dia dos alunos, fazendo-se necessário que sejam ensinados os conceitos fundamentais da computação. Com a inserção do ensino de computação na educação básica, espera-se que o aluno aprenda mais que o simples uso da tecnologia. Isso permitiria desenvolver o pensamento computacional com o objetivo de solucionar problemas a partir de um conjunto de conceitos, como: abstração, recursão, iteração, etc (TUCKER, 2003).

Valente (1993) afirma que para alguns educadores, a utilização de computadores na escola pode causar déficit de aprendizagem. Os argumentos são que escola, professores e alunos, teriam dificuldades em se adaptar a uma nova abordagem educacional. Valente (1993) ainda afirma que isso se dá por receio e incerteza do que não vivenciaram. Mitos e barreiras como essas podem ser quebradas com a junção dos conceitos de Ciência da Computação e estratégias de ensino (NAPOLITANO; BATISTA, 2003).

O raciocínio lógico de crianças que aprendem a computação nas escolas é geralmente melhor quando comparado a crianças que não passam por esse aprendizado. Para Napolitano e Batista (2003) o uso adequado dos computadores desperta a curiosidade das crianças, fazendo com que elas busquem soluções por conta própria. Outras vantagens que podem ser citadas são: interação entre aluno e professor para resolver problemas; estímulo do pensamento lógico e crítico; incentivo à pesquisa e à criatividade; motivação à utilização de computadores como meio de aprendizagem (MATTEI, 2011).

3 ALGORITMOS E SUAS FORMAS DE REPRESENTAÇÃO

A proposta da Base Nacional Comum Curricular (2016) define tecnologias digitais como projeto integrador nas escolas. No entanto, concentra-se mais no uso das tecnologias do que na sua produção. Para que os alunos sejam capazes de criar seus próprios jogos e programas/aplicativos, é preciso entender uma nova linguagem que permite esse desenvolvimento.

Um programa de computador funciona seguindo instruções contidas no seu interior. Cada instrução do programa depende de outras instruções, formando assim blocos de instruções. Esses blocos devem ser encadeados de forma a compor uma lógica que resolva um problema. Esse encadeamento das instruções é a lógica de programação, também conhecida como algoritmo.

Um algoritmo é composto por procedimentos realizados passo a passo para resolver uma tarefa. Quando um cozinheiro prepara um bolo seguindo as instruções de uma receita, por exemplo, ele está seguindo um algoritmo. De forma um pouco mais técnica, pode ser definido como uma sequência de comandos lógicos que dão instruções ao computador/dispositivo para resolver um problema ou executar uma tarefa (CORMEN, 2009).

As pessoas utilizam algoritmos o tempo todo, porém de forma intuitiva e automática. A cada atividade simples realizada, existe um algoritmo determinado que pode passar despercebido. Um exemplo de algoritmo rotineiro para preparar um bolo é apresentado na Figura 1.

Figura 1 – Exemplo de algoritmo para preparar um bolo usando lista de comandos

Ingredientes:

- 2 xícaras de açúcar;
- 3 xícaras de farinha de trigo;
- 4 colheres de margarina;
- 3 ovos;
- 1 e 1/2 xícara de leite;
- 1 colher (sopa) bem cheia de fermento em pó.

Modo de preparo:

1. Pré-aqueça o forno a 180 graus;
1. Em um recipiente, bata as claras de ovos até ficar em ponto de neve;
2. Em outro recipiente, bata a farinha, açúcar, margarina, leite e as gemas até ficar homogêneo.

Em seguida, acrescente o fermento;

3. Adicione na massa as claras em neve e mexa;
4. Despeje a massa em uma forma untada;
5. Coloque a forma no forno;
6. Deixe assando por 35 minutos;
7. Retire o bolo do forno e espere esfriar;
8. Desenforme e sirva.

Fonte: Próprio autor.

No exemplo apresentado na Figura 1 são mostrados os ingredientes e os passos

para se fazer um bolo. A receita é apresentada na forma de uma lista de instruções, algo muito conhecido dos cozinheiros. Nota-se que cada linha descreve uma instrução ou passo a ser seguido. É muito importante que cada passo seja executado corretamente e na ordem em que são apresentados. Se esse critério não for seguido à risca, o resultado final pode não ser o esperado.

Fazendo um paralelo com linguagem de programação, os ingredientes são entendidos como as variáveis do problema. As instruções de como manipular os ingredientes correspondem aos comandos que manipularão as variáveis. Uma forma de se apresentar um algoritmo, independente de uma linguagem de programação, é na forma de um pseudocódigo. Para Cormen (2009) o pseudocódigo estrutura os passos de um algoritmo de maneira lógica e de fácil entendimento. Na Figura 2 é apresentada a receita descrita na Figura 1 na forma de um pseudocódigo.

Figura 2 – Exemplo de algoritmo para preparar um bolo usando lista de comandos

Variáveis

```
acucar = 2
farinha = 3
margarina = 4
clara_ovo = 3
gema_ovo = 3
leite = 1.5
fermento = 1
temperatura = 180
ponto_de_neve = falso
mistura_homogenea = falso
tempo = 35
recipiente1 = 0
recipiente2 = 0
forma = 0
bolo = 0
```

Início

```
aquecer_forno(temperatura)
colocar_no_recepieinte(recepieinte1, clara_ovo)
enquanto(ponto_de_neve == falso) faça
  ponto_de_neve = bater_claras(recepieinte1)
fim-enquanto
colocar_no_recepieinte(recepieinte2, farinha, gema_ovo, acucar, margarina, leite)
  enquanto(mistura_homogenea == falso) faça
    mistura_homogenea = bater_ingredientes(recepieinte2)
  fim-enquanto
colocar_no_recepieinte(recepieinte2, fermento)
  forma = recipiente1 + recipiente2
colocar_no_forno(forma,tempo)
bolo = retirar_do_forno(forma)
servir(bolo)
```

Fim

Fonte: Próprio autor.

Um aplicativo é desenvolvido utilizando-se uma linguagem de programação. Existem

inúmeras linguagens de programação tais como C (DEITEL *et al.*, 2011), C++ (DEITEL; DEITEL, 2006), C# (DEITEL; DEITEL, 2003), Java (DEITEL; DEITEL, 2005), PHP (DAVIS; PHILLIPS, 2008), Python (MENEZES, 2010) e muitas outras. Apesar de cada linguagem ter uma sintaxe específica, todas elas apresentam os conceitos dos blocos de comando do pseudocódigo. Ou seja, um algoritmo é independente da linguagem de programação utilizada para codificá-lo. Em outras palavras, um pseudocódigo que descreve uma lógica, pode ser traduzido para qualquer linguagem de programação. A Figura 3 apresenta a tradução do pseudocódigo da Figura 2 para a linguagem PHP.

Figura 3 – Exemplo de algoritmo para preparar um bolo usando lista de comandos

```
<?php
    $acucar = 2;
    $farinha = 3;
    $margarina = 4;
    $clara_ovo = 3;
    $gema_ovo = 3;
    $leite = 1.5;
    $fermento = 1;
    $temperatura = 180;
    $ponto_de_neve = false;
    $mistura_homogenea = false;
    $tempo = 35;
    $recepte1 = 0;
    $recepte2 = 0;
    $forma = 0;
    $bolo = 0;

    aquecer_forno($temperatura);
    colocar_no_recepte($recepte1, $clara_ovo);
    while($ponto_de_neve == false){
        $ponto_de_neve = bater_claras($recepte1);
    }
    colocar_no_recepte($recepte2, $farinha, $gema_ovo, $acucar, $margarina, $leite);
    while($mistura_homogenea == false){
        $mistura_homogenea = bater_ingredientes($recepte2);
    }
    colocar_no_forno($forma, $fermento);
    $forma = $recepte1 + $recepte2;
    colocar_no_forno($forma, $tempo);
    $bolo = retirar_do_forno($forma);
    servir($bolo);
?>
```

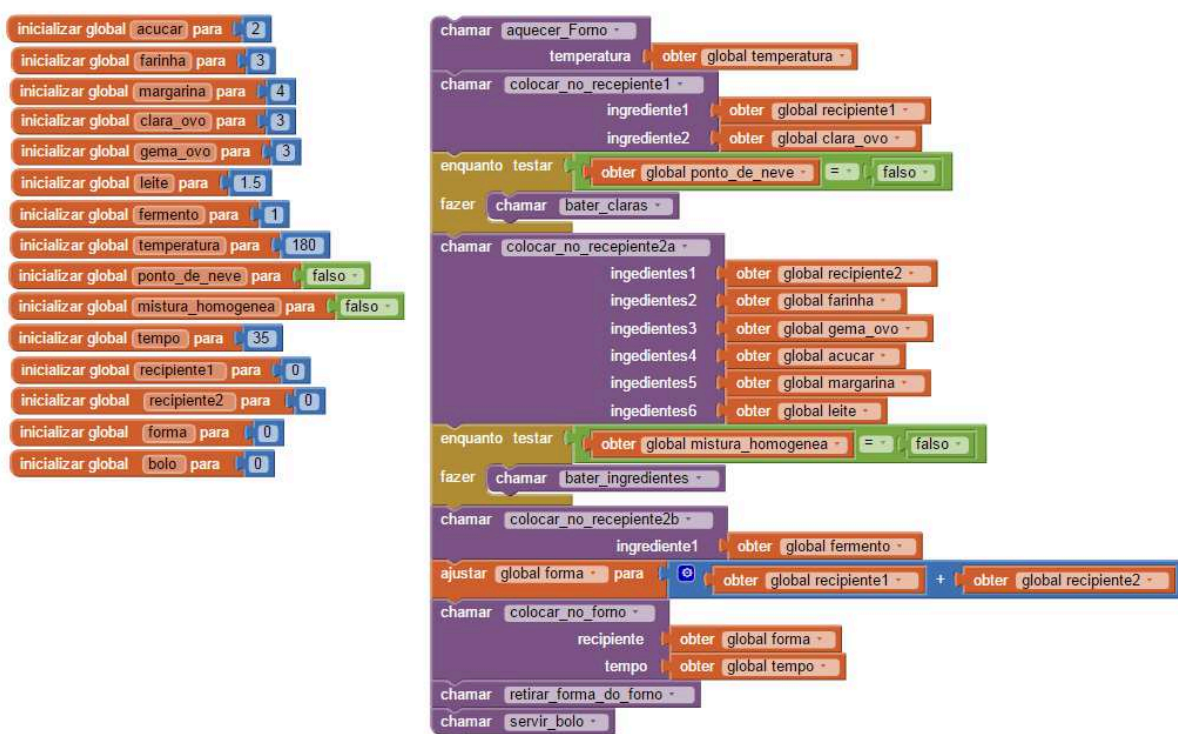
Fonte: Próprio autor.

Pensando no ensino de programação para alunos do ensino básico, o entendimento dos pseudocódigos e das linguagens de programação pode ser algo complexo e entediante. Um dos desafios é como ensinar de forma agradável e prazerosa. Para isso, existem muitos

recursos computacionais que podem ser usados como método de ensino, sem utilizar uma linguagem de programação. Cada ferramenta apresenta um formato diferente, como: jogos digitais, programas/aplicativos educacionais, plataformas digitais e etc. Outra forma de ensinar é com ferramentas que usam linguagem de blocos. Essa forma consiste em manusear os comandos de forma gráfica, encaixando-os de acordo com a lógica de programação, como se fossem peças de um quebra-cabeça. Essas ferramentas além de serem de fácil de interação, apresentam recursos dinâmicos e de fácil entendimento.

Na Figura 4 é mostrada a mesma tarefa de fazer um bolo apresentado na Figura 1. O algoritmo é representado na forma de blocos encaixados. Observa-se que a lógica apresentada na Figura 4 é equivalente à das Figuras 2 e 3, porém de uma maneira muito mais intuitiva e visual. Isso a torna mais simples de entender e mais atrativa, sendo uma forma adequada de apresentar os princípios de lógica e programação para o ensino básico.

Figura 4 – Exemplo de algoritmo para preparar um bolo usando programação em blocos



Fonte: Próprio autor.

Na próxima seção serão apresentadas ferramentas que usam a linguagem de blocos tanto para ensinar os conceitos de lógica e programação, quanto para o desenvolvimento de aplicativos reais. Para exemplificar serão detalhadas duas ferramentas. Ao final será mostrado um passo a passo do desenvolvimento de um aplicativo para celular.

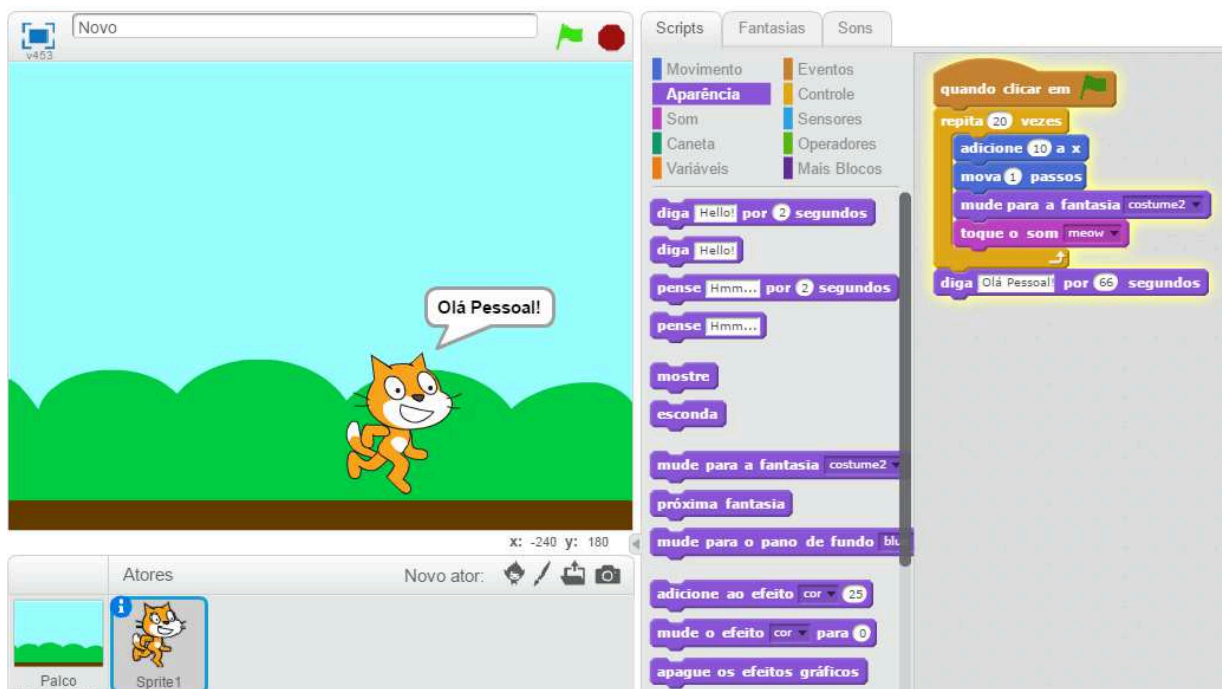
4 FERRAMENTAS DE ENSINO DE LÓGICA E PROGRAMAÇÃO UTILIZANDO BLOCOS

Existem várias ferramentas de ensino de programação que utilizam a metodologia de blocos. Segundo Feitosa (2001), elas têm como característica predominante o fato de facilitar a aprendizagem dos alunos. O professor poderá escolher a ferramenta que melhor relaciona a tecnologia à sua proposta educacional. Na Tabela 1 são apresentadas algumas delas, bem como os desenvolvedores e as principais características de cada uma. Com tantas opções disponíveis, cabe ao professor e a escola escolherem as ferramentas que melhor se adaptam a idade dos alunos. Muitas disponibilizam a faixa de idade recomendada.

As ferramentas apresentadas na Tabela 1 possuem algumas características semelhantes: são voltadas para crianças; utilizam o método de blocos; são interativas. Pela quantidade de recursos disponíveis, pode-se trabalhar em conjunto com outras disciplinas escolares. Pode-se citar como exemplo de aplicação interdisciplinar: ângulos e geometria em matemática; vetores em física; geolocalização em geografia; interpretação de texto em português; estudo da língua inglesa; etc.

Um exemplo de ferramenta, apresentada na Tabela 1, é o *Scratch*, ilustrado na Figura 5. Conforme Resnick *et al.* (2009) a proposta do *Scratch* é ensinar lógica de programação para crianças e iniciantes, utilizando recursos gráficos que permitem criação de jogos e animações simples.

Figura 5 – Scratch, ferramenta de ensino de lógica de programação para crianças



Fonte: Scratch (2016). Adaptado.

Tabela 1 – Ferramentas de ensino de lógica de programação que utilizam metodologia de blocos

| Ferramenta | Ícone / Logotipo | Desenvolvedor | Grau de dificuldade | Características |
|----------------|---|--------------------------|-----------------------|--|
| Allcancode |  | Allcancode Inc | Muito baixo | Jogos para crianças com animações interativas. |
| App Inventor |  | Google e MIT | Médio | Ferramenta voltada para o desenvolvimento de aplicativos Android. |
| Blockly Games |  | Google | Muito baixo até Médio | Utilização de lógica para resolução de problemas em formato de jogos. |
| Code |  | Organização filantrópica | Muito baixo até médio | Lógica de programação para crianças em idade de alfabetização. |
| Codebug |  | Openlx | Médio | Cria programas para dispositivos interativos. Necessita de <i>hardware</i> específico. |
| Disney Jogos |  | Disney | Muito baixo | Jogos com muita interatividade que utilizam lógica de programação para crianças. |
| Gameblox |  | MIT | Alto | Editor de criação de jogos. |
| Gamefroot |  | GameFroot | Muito Alto | Cria jogos, permite compartilhamento, podendo até mesmo disponibilizar para venda. |
| Leamtomod |  | Multi-Dimensional Games | Alto | Criação de módulos para o jogo Minecraft. Ferramenta paga. |
| Made With Code |  | Google | Alto | Animações baseadas em blocos. |
| Micro Bit |  | BBC | Médio | Possibilita a construção de programas interativos que podem ser executados em microcomputadores. |
| Open Roberta |  | Google e LEGO | Baixo | Ensina a programar robôs. |
| Ozo Blockly |  | Ozobot & OzoBlockly | Médio | Cria códigos para animação de dispositivos Ozobot (o robô). |
| Scratch |  | MIT | Muito Baixo | Cria histórias, jogos e animações. Permite compartilhar com pessoas de todo o mundo. |

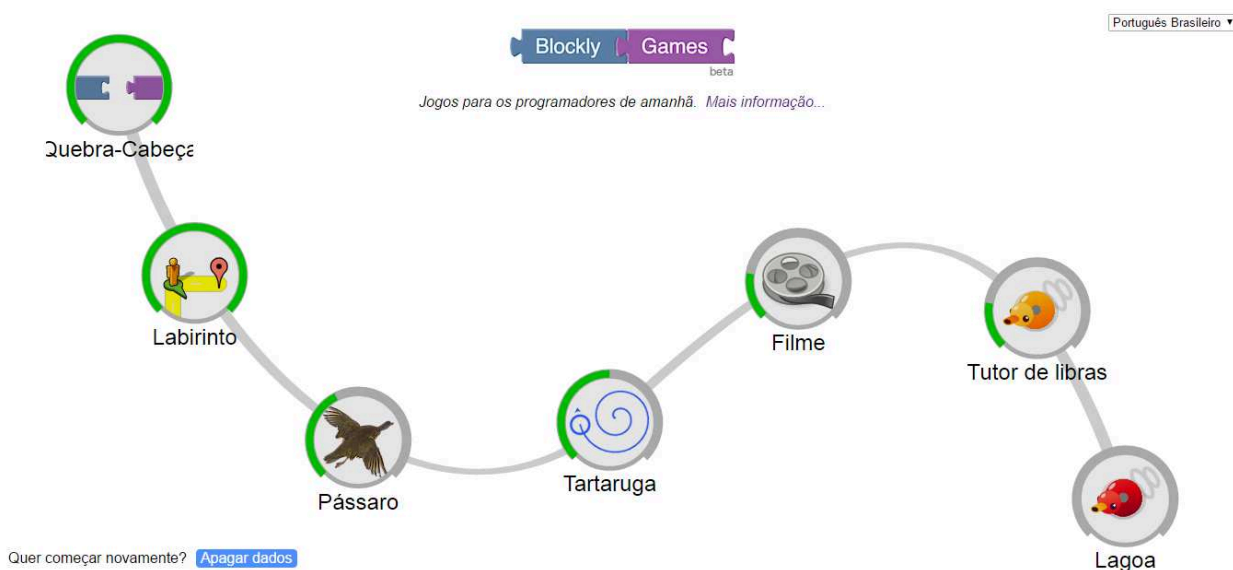
Observa-se na Figura 5 que existem alguns blocos com funções específicas localizados no centro da tela. Eles podem ser arrastados até o lado mais à direita e encaixados de forma a montar ações para que o personagem da parte mais à esquerda da tela as execute. O *Scratch* funciona também como uma grande comunidade de desenvolvimento, onde as crianças podem compartilhar suas criações com o mundo todo. Scaico *et al.* (2013) discutem o ensino de programação em escolas públicas do ensino médio utilizando como ferramenta o *Scratch*. Nesse trabalho os autores detalham a ferramenta e como foi utilizada.

A seguir será apresentado com detalhadamente outra ferramenta baseada em blocos, o *Blockly*. Ela não tem como características o desenvolvimento de programas/aplicativos, mas de introduzir lógica de programação através de jogos digitais, de maneira simples, intuitiva e divertida.

4.1 *Blockly Games*

O *Blockly Games* é uma série de jogos educativos que visam ao ensino de lógica de programação. Ele foi desenvolvido para crianças que não tiveram experiência prévia com nenhuma linguagem de programação. O *Blockly* é intuitivo, o que permite que o usuário descubra a ferramenta de forma independente. Todos os jogos podem ser acessados na mesma página sem ser necessário fazer um registro no site. Além de ser uma ferramenta que não requer grandes recursos computacionais, funciona em *tablets* e celulares. O *Blockly* também pode ser descarregado para o computador, dispensando o uso de *Internet* para funcionar, necessitando apenas de um navegador instalado (BLOCKLY GAMES, 2016). Essa ferramenta contém sete jogos, como mostrado na Figura 6.

Figura 6 – Tela inicial do *Blockly Games* com os sete jogos disponíveis

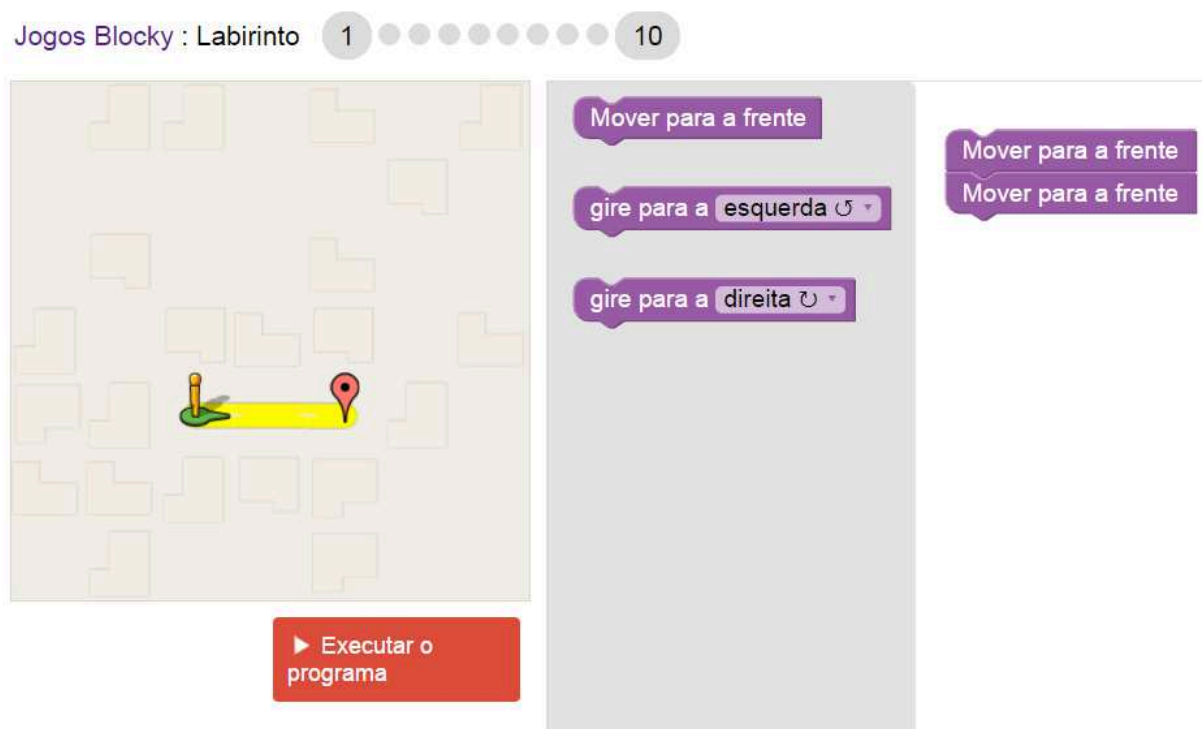


Fonte: Blockly Games (2016). Adaptado.

O propósito é resolver um determinado exercício enviando comandos através da montagem dos blocos. De forma geral, os jogos possuem fases com níveis de dificuldade gradativos. A cada novo desafio, aumenta-se o nível de dificuldade, apresentando-se novos blocos. Caso o usuário não consiga resolver um determinado exercício, pode-se optar por pular aquele nível, podendo voltar a qualquer momento. Isso faz com que o usuário não se canse do exercício e não desista no primeiro obstáculo.

Na Figura 7 é mostrado o nível 1 do jogo Labirinto. Nota-se que ao centro estão localizados os blocos que podem ser usados, cada qual com uma função específica. À esquerda da figura está localizado o exercício proposto. Para resolver o exercício, deve-se mover os blocos para o lado mais à direita, encaixando-os de forma lógica.

Figura 7 – *Blockly Games*: Jogo do labirinto



Fonte: Blockly Games (2016). Adaptado.

Cada bloco de comando representa uma ação do personagem. No exemplo da Figura 7, quando o programa for executado, o personagem caminhará duas vezes para frente, como definido na montagem dos blocos mais à direita. É um exercício bem simples, contudo importante para que o aluno entenda como funciona a ferramenta. Em alguns níveis será necessário que se use blocos de repetição ou laço. Na Figura 8 é mostrado um exemplo do uso do bloco de repetição.

Na Figura 8, nível 4 do jogo, existe um limite de 5 blocos para resolver o exercício. Nota-se que existe um novo bloco que tem a função de repetir outros comandos. O bloco “repita até ... faça” executará até que o personagem chegue ao destino, economizando o número de blocos usados. Sem esse bloco seriam necessários 15 passos para concluir esse nível.

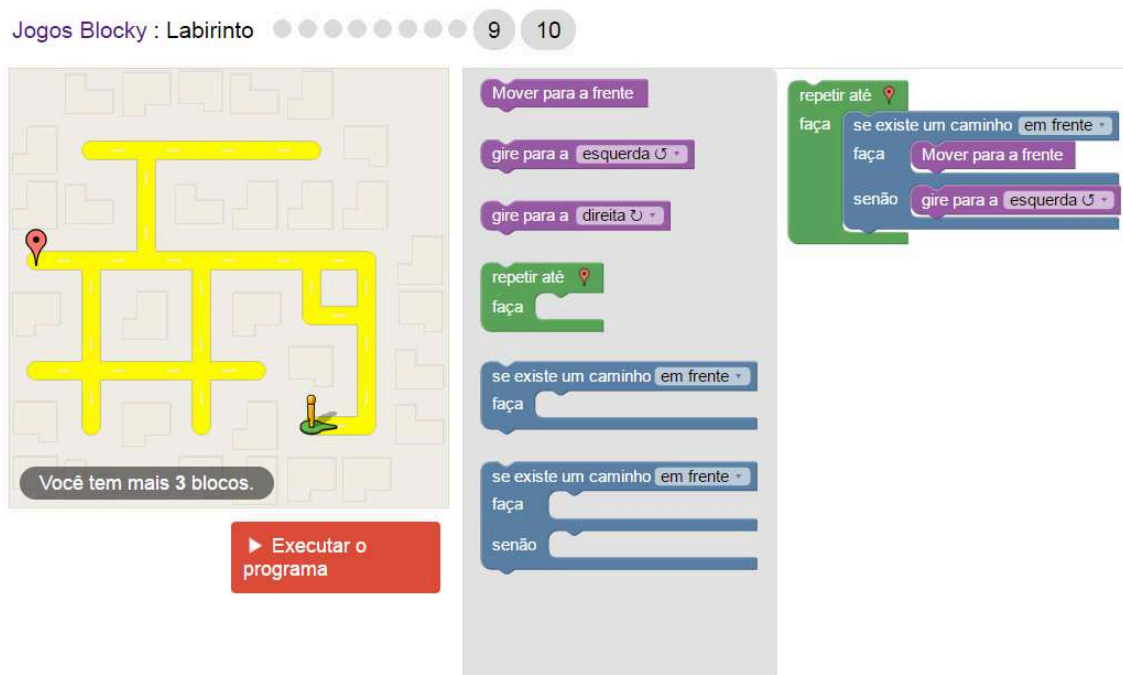
Figura 8 – Uso de bloco de repetição



Fonte: Blockly Games (2016). Adaptado.

Neste exercício há apenas um caminho para o personagem seguir, não há escolhas a se fazer nele. Em muitos casos é necessário tomar uma decisão baseado em alternativas, como mostrado na Figura 9.

Figura 9 – Uso de bloco condicional

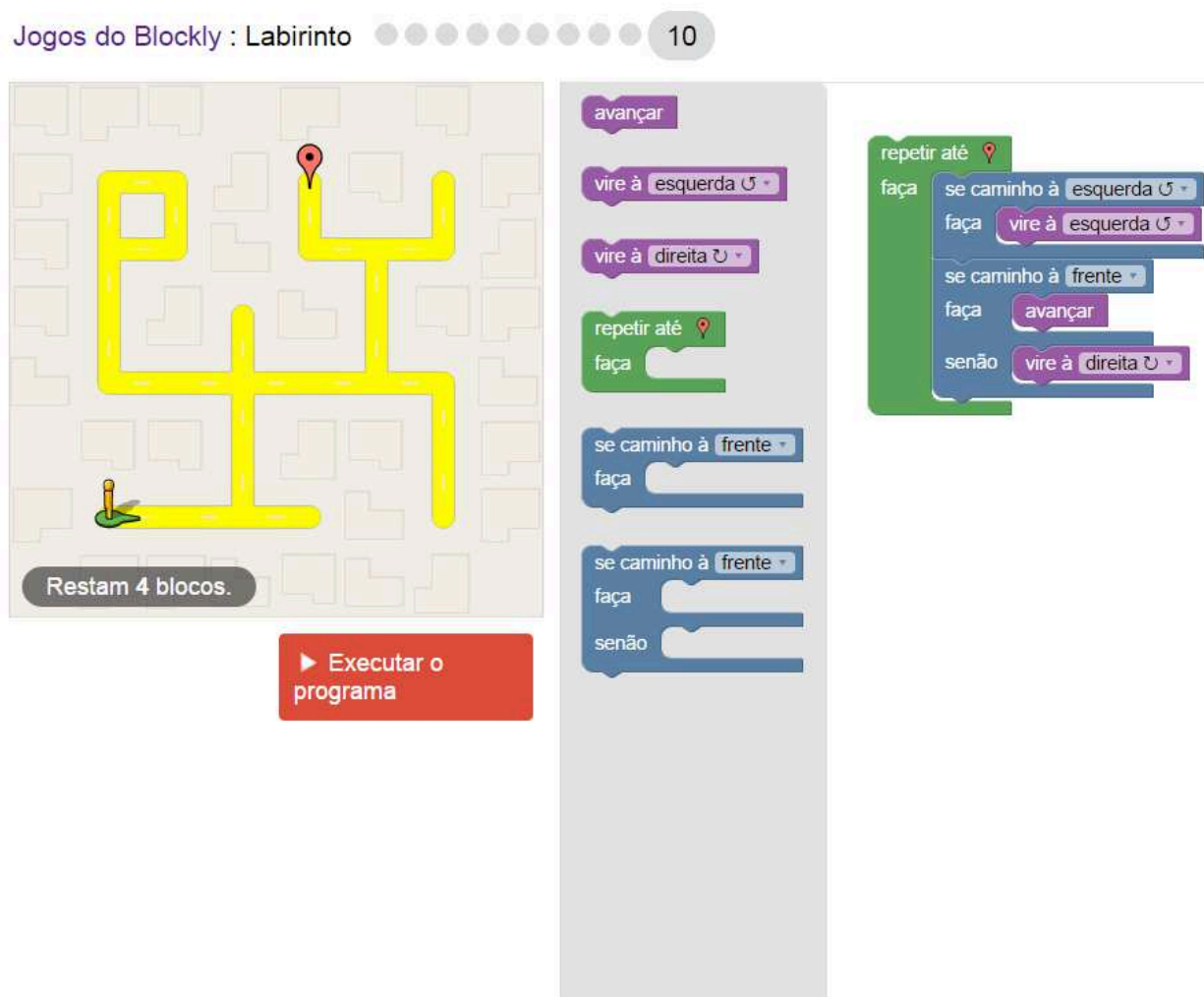


Fonte: Blockly Games (2016). Adaptado.

Na Figura 9 é apresentado o nível 9 do labirinto. Nela fica visível que o personagem em alguns momentos pode seguir por caminhos diferentes. Nesse nível o usuário precisará usar uma estrutura condicional. Observa-se que no bloco condicional em azul, o personagem apenas se moverá para frente caso exista um caminho livre (faça), caso contrário outra condição será executada (senão).

A solução do nível 9 é uma solução específica para o problema proposto. Em um labirinto diferente o mesmo algoritmo não resolveria o exercício. Na implementação de lógica de programação se propõem criar algoritmos genéricos que resolvam qualquer situação de uma determinada família de problemas. Na Figura 10 é mostrado um problema onde a solução criada resolveria qualquer tipo de labirinto.

Figura 10 – Algoritmo que solucionaria qualquer labirinto



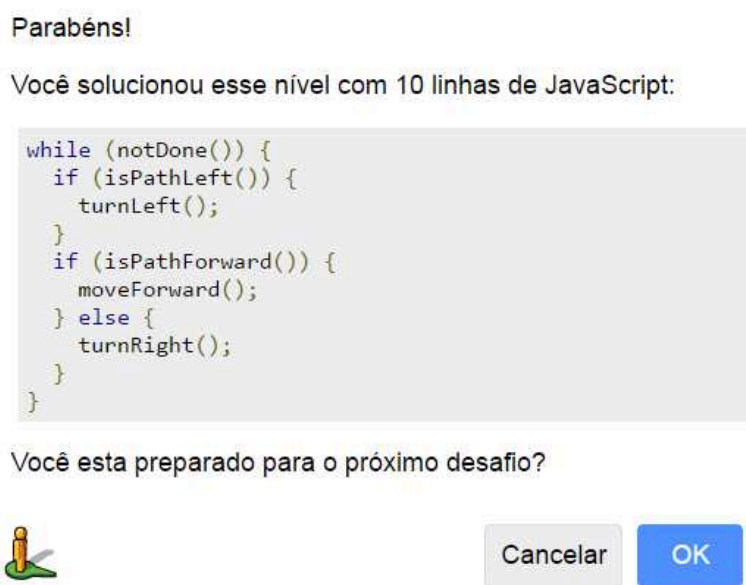
Fonte: Blockly Games (2016). Adaptado.

Vale destacar que a solução da Figura 10 não é a única possível, sendo que o usuário poderá encontrar soluções diferentes para o mesmo problema. Na solução apresentada o personagem irá virar para esquerda sempre que houver um caminho à esquerda, depois ele irá

verificar se há um caminho para frente. Se sim ele avança, senão ele vira à direita. O algoritmo repetirá essas ações até o personagem chegar ao destino determinado. Ao executar o algoritmo nota-se que o personagem não irá seguir diretamente até o destino. A intenção dele não é otimizar uma solução específica, mas resolver qualquer labirinto.

Todos os blocos do jogo são representações de comandos reais usados em linguagens de programação. Na forma como é apresentado no *Blockly*, o entendimento dos conceitos primordiais da programação é facilitado. Ao final de cada nível, o jogo mostra o código em linguagem JavaScript (FLANAGAN, 2012), como ilustrado na Figura 11. Isso permite que o usuário tenha uma visão de como seria o código em uma linguagem de programação textual.

Figura 11 – Código em linguagem JavaScript correspondente ao algoritmo da Figura 10



Fonte: Blockly Games (2016). Adaptado.

O *Blockly* não resulta em um programa real. Existem outras ferramentas onde os usuários podem criar e testar seus próprios programas/aplicativos. Pode-se citar como exemplo o *App Inventor*. A seguir será apresentada essa ferramenta e será mostrado também como desenvolver uma aplicação real para Android usando metodologia em blocos.

4.2 Ambiente *App Inventor*

O *App Inventor* é um ambiente de desenvolvimento de aplicativos baseado em blocos. Conforme Wolber (2011) ele foi desenvolvido pela Google juntamente com o MIT com o intuito de incentivar o desenvolvimento de aplicações para dispositivos com sistema Android. Assim como no *Blockly Games*, o *App Inventor* utiliza os blocos para criar uma solução. Entretanto, o programa criado pode utilizar recursos de *hardware* do dispositivo, além de incorporar serviços baseados na *web*. Como exemplos de recursos têm-se: leitura de código de barras; câmera

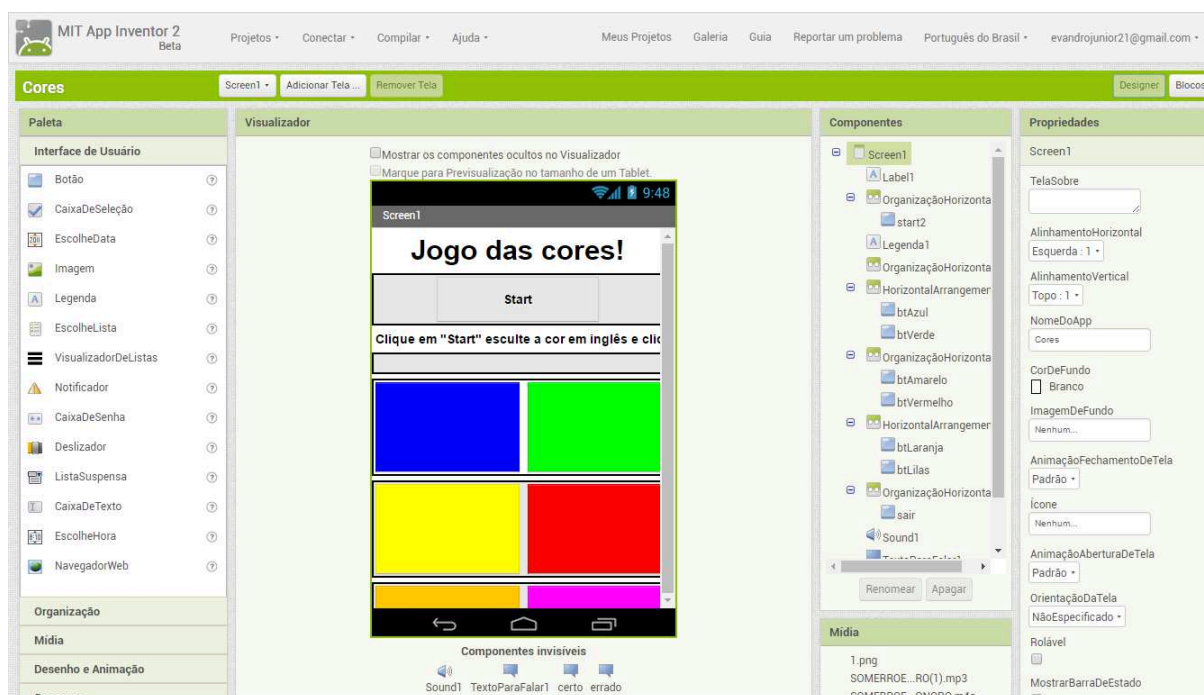
do celular; geolocalização; integração com redes sociais; reconhecimento de fala; etc. O *App Inventor* está disponível em vários idiomas, dentre eles o português do Brasil.

O ambiente permite ao usuário criar suas aplicações e desenvolver a criatividade através da interação com a ferramenta. Um dos propósitos é tornar o aprendizado atraente e intuitivo, além de fornecer um método para criação de aplicativos avançados mesmo sem o domínio pleno de programação (ABELSON, 2009).

Para utilizar o ambiente é necessário que o usuário tenha uma conta no Google. A área de trabalho do *App Inventor* é dividida em design e blocos. Na parte de *design* pode-se montar a interface do aplicativo apenas arrastando os componentes e soltando no desenho da tela do celular. Na parte de blocos, é feita a organização das ações dos componentes e as interações com o usuário.

Na Figura 12 é mostrado um exemplo de interface de um aplicativo criado no *App Inventor*. Ele consiste em um jogo educativo com o propósito de ensinar as cores em inglês para crianças.

Figura 12 – *App Inventor*: Modo de design



Fonte: MIT APP Inventor 2 (2016). Adaptado.

Nesse aplicativo foram utilizados recursos básicos da ferramenta como textos, botões e organizadores. Esses componentes podem ser alterados pelo usuário na aba propriedades. Nela é possível mudar a cor do objeto, tamanho, alinhamento, fonte, margens e etc. Com a criatividade do desenvolvedor e os variados recursos disponíveis no *App Inventor* é possível criar uma grande variedade de interfaces. No exemplo, o jogador deve clicar em *start* para ouvir um nome de uma cor em inglês e em seguida clicar no quadrado com a cor correspondente. Na Figura 13 é apresentada a interface do aplicativo sendo executado em um celular.

Figura 13 – Interface do aplicativo de cores executado em um celular com sistema Android



Fonte: Próprio autor.

Além de criar o design do aplicativo, o usuário deve dar funcionalidades aos componentes. Para isso são usados blocos. Um exemplo de lógica de programação a partir da montagem dos blocos é apresentado na Figura 14.

Figura 14 – *App Inventor*: Modo de desenvolvimento de lógica de programação utilizando blocos



Fonte: MIT APP Inventor 2 (2016). Adaptado.

No exemplo foram criadas duas variáveis: uma numérica com o nome *numero_cor* inicializada com o valor 0; e uma do tipo texto com o nome *cor* inicializada como vazia. Quando

o usuário clicar no botão *start* a variável *numero_cor* receberá um valor aleatório de 1 a 6. No bloco condicional se do *start*, de acordo com o valor sorteado, será determinado o texto para variável *cor*, que é uma das cores disponíveis no jogo. Essas variáveis serão utilizadas nos demais blocos para verificar se o usuário clicou no quadrado correspondente à cor sorteada. Se o usuário clicar na cor correspondente à cor sorteada, o aplicativo dará os parabéns pelo acerto. Caso contrário, o aplicativo pedirá que o usuário tente novamente.

Na página do *App Inventor* são disponibilizados vários tutoriais e exemplos de aplicativos criados por professores e alunos, bem como relatos de experiência de uso em sala de aula. Inicialmente, essa ferramenta foi utilizada em projetos-piloto para introduzir programação em escolas de ensino médio dos Estados Unidos. Hoje tem sido utilizada em diversas instituições de ensino (MIT APP INVENTOR 2, 2016). Estudos mostram que os alunos se sentiram muito mais motivados em aprender programação utilizando o *App Inventor* e que os resultados alcançados pelos professores foram satisfatórios (ABELSON, 2009; BEGHINI, 2013).

Verifica-se assim que o *App Inventor* tem grande apelo pedagógico. Ele pode ser utilizado como um fator diferencial de ensino nas escolas, visando ao desenvolvimento de pensamento computacional e crítico. Para tanto, é necessário que sua aplicabilidade seja uma realidade nas escolas básicas do Brasil, viabilizando a aprendizagem dos alunos e a interdisciplinaridade escolar.

5 LÓGICA DE PROGRAMAÇÃO NA EDUCAÇÃO BÁSICA

As disciplinas ensinadas nas escolas visam a fornecer conteúdo aos alunos de uma determinada área de conhecimento. Pensando nisso, é importante que os alunos estejam preparados para resolver problemas e tomar decisões, baseando-se em conceitos aprendidos na própria escola. A lógica de programação engloba muitos conceitos que podem ser amplamente aplicados em diversas situações.

A Sociedade Brasileira de Computação (SBC) é uma das grandes incentivadoras da proposta de ensino de lógica de programação na educação básica. Ela recomenda que o ensino seja a partir do ensino básico para que os princípios e habilidades ensinadas contribuam com a resolução de problemas (BARCELOS; SILVEIRA, 2012).

O ensino de lógica de programação na educação básica pode ajudar os alunos a se orientarem sobre carreiras profissionais. No trabalho realizado por Scaico *et al.* (2012) foi utilizado o *Scratch* para demonstrar que os alunos utilizaram os conceitos aprendidos para entender o que acontece em um curso superior da área de computação. Com tantos conceitos prévios, que seriam aprendidos nas escolas, os alunos de cursos de graduação estariam mais preparados para lidar com as dificuldades iniciais. Para Barcelos e Silveira (2012) e Melo, Costa e Batista (2013), isso promoveria um menor índice de desistência ou reprovação em cursos afins. Esses motivos mostram a relevância dessa proposta voltada para crianças e jovens, principalmente pelo fato de estarem em processo de maturação do conhecimento.

Acreditar que a lógica de programação pode ser uma disciplina escolar, implica em apresentar fundamentos suficientes para que isso se torne significativo aos alunos. Para tanto, o trabalho intitulado “*Computer science: a curriculum for schools*”, escrito por colaboradores de grandes empresas, como Google® e Microsoft®, faz essa tentativa. Segundo os autores, a Ciência da Computação está profundamente preocupada como os computadores e sistemas são projetados. Os alunos que estudam computação obtêm ganhos de todos os tipos, ligados ou não com os computadores. A lógica de programação influencia nas diversas áreas de ensino, como na biologia, química, linguística, psicologia, economia e estatística. A computação permite ao aluno compreender um determinado problema e criar um método para solucioná-lo. Com a programação pode-se entender melhor os limites da inteligência humana e das máquinas. Eles também afirmam que o aluno com pensamento computacional está mais preparado para compreender o computador, assim pode-se extrair mais dele e aplicar para sociedade (COMPUTING AT SCHOOL WORKING GROUP, 2012).

A escolha de uma ferramenta apropriada é essencial nesse contexto. Wolber (2011) descreve que na realização de um curso introdutório sobre *App Inventor* os alunos criaram aplicações práticas para o cotidiano, das quais podem ser citadas: “*Android Where’s My Car*”⁶,

⁶ <http://appinventor.mit.edu/explore/ai2/android-wheres-my-car.html>

aplicativo para localização de onde o carro foi estacionado; e o “*No Text While Driving*”⁷ aplicativo que avisa com mensagem de texto que o motorista está dirigindo. Nesse curso, Wolber (2011) relata que os alunos se mostraram muito mais motivados e demonstraram aprender mais o conteúdo que foi proposto através da lógica de programação. Em um estudo quanti-qualitativo, realizado em uma turma de alunos do ensino básico do país, percebeu-se que os alunos foram capazes de ampliar o raciocínio lógico através do *Blockly*. A ferramenta foi capaz de estimular o aprendizado dos alunos. Mostrou-se também a necessidade que os alunos têm de serem estimulados de forma lúdica (BLOCKLY GAMES, 2016).

De acordo com as ferramentas apresentadas na Tabela 1, o professor pode escolher uma ou mais para desenvolver atividades na sala de aula. Algumas dessas ferramentas proporcionam diversão e aprendizagem mais voltada para crianças. Essas ferramentas podem ser aplicadas para alunos do ensino fundamental. Ferramentas mais sofisticadas, como o *App Inventor*, podem ser aplicadas tanto no ensino básico quanto nos ensinos médio e superior.

Neste trabalho sugere-se que o desenvolvimento de atividades para ensinar lógica de programação em sala de aula siga os seguintes princípios:

- a) **Ensinar um conteúdo:** é importante que ao mesmo tempo que o professor trabalhe o raciocínio lógico do aluno um conteúdo curricular seja repassado ao mesmo;
- b) **Manter o aluno motivado:** se a atividade não for levada até um determinado propósito o aluno pode se sentir frustrado. Toda atividade deve ter um início, meio e fim e a motivação é importante para que isso ocorra;
- c) **O ensino de lógica de programação deve provocar diversão:** a proposta de ensinar programação usando blocos é promover o aprendizado de forma prazerosa para o aluno.

A partir do estudo realizado e das experiências de outros países é mostrado que a lógica de programação tem grande potencial pedagógico. A proposta é válida e visa a contribuir com a aprendizagem interdisciplinar e o desenvolvimento do raciocínio lógico dos alunos.

⁷ <http://appinventor.mit.edu/explore/no-text-while-driving-ai2.html>

6 INTERDISCIPLINARIDADE DA LÓGICA DE PROGRAMAÇÃO

Interdisciplinaridade é a interação entre disciplinas com o intuito de trocar conhecimento de forma recíproca (PIAGET; BARROS, 1973). Para Barbier (2002), a interdisciplinaridade provoca parceria entre disciplinas, formando uma articulação horizontal entre conceitos e aprofundamento vertical no conhecimento de cada disciplina. Uma das características fundamentais da prática interdisciplinar é a experiência que ela proporciona em adquirir conhecimento específico utilizando-se de outra ciência. Contudo, Ferreira (2005) alerta que a intenção do trabalho interdisciplinar deve ser clara e objetiva por parte de quem a pratica. Morin (2014) acredita que o ensino formal das escolas básicas dificulta o processo de aprendizagem dos alunos, não estimula o desenvolvimento da inteligência e prejudica a capacidade de estabelecer ligação entre os conceitos. Ele ainda completa afirmando que as disciplinas apresentadas de forma tradicionalistas geram ao aluno uma inteligência parcelada, mecanicista e reducionista. No processo de aprendizagem interdisciplinar o aluno passa a assumir um papel diferente dentro da escola, não sendo mais somente um espectador, mas um construtor de conhecimento. Seguindo todos estes conceitos depara-se com um novo paradigma educacional. No Brasil, com a nova proposta de reformulação da base curricular, abre-se uma porta para uma reestruturação curricular, metodológica e funcional. Conforme a LDB (Lei de Diretrizes e Bases da Educação) prevê, o ensino educacional deve implementar a interdisciplinaridade dos conteúdos curriculares (LDB, 2015).

A partir da década de 50, com o surgimento do Parque Científico e Tecnológico nos Estados Unidos (origem do Vale do Silício), a inovação se tornou um fator importante no desenvolvimento econômico e social do mundo. A computação foi o fator primordial para este desenvolvimento, em especial pelo fato de atuar em várias áreas de conhecimento (VICARI, 2016). A educação vem progredindo através de variados meios e instrumentos procurando oferecer recursos didáticos lúdicos. A tecnologia educacional, procedimentos, meios e equipamentos usados nas escolas, são adotados com a finalidade de facilitar o processo de ensino e aprendizagem. No atual cenário tecnológico torna-se importante que a computação e seus sistemas sejam protagonistas na metodologia interdisciplinar também nas escolas. Sendo assim, há a necessidade de buscar formas de uso para as ferramentas que a computação disponibiliza.

Em um trabalho realizado por Westphalen, Toni e Santos (2000) em uma escola estadual do Paraná, a informática foi implantada como disciplina instrumental no ensino médio. Nessa experiência foram utilizadas ferramentas do computador para desenvolver o conhecimento científico e investigativo dos alunos. Com a aplicação desse projeto percebeu-se uma mudança na forma com que os mesmos resolviam problemas de outras áreas. Notou-se também que eles se tornaram mais críticos após o uso das ferramentas (WESTPHALEN; TONI; SANTOS, 2000).

A internet é um campo repleto de informações e a qualidade dessas torna-se algo essencial na escolha das ferramentas que serão utilizadas nas escolas. Para tanto, na Seção 7

deste trabalho é proposto um *website* ou plataforma que forneça esse apoio informativo aos professores e alunos sobre as ferramentas de ensino de lógica de programação.

7 PLATAFORMA DE SUPORTE ÀS FERRAMENTAS DE ENSINO DE LÓGICA DE PROGRAMAÇÃO

A utilização de ferramentas didáticas em sala de aula possibilita que o processo de ensino e aprendizagem seja dinâmico, interativo e contextualizado à realidade atual dos alunos. No entanto, é preciso ter cuidado com o uso da tecnologia como ferramenta pedagógica para que não haja uma aplicação de forma inadequada. A tecnologia pode contribuir para obter maior atenção dos alunos, consequentemente o uso coerente das ferramentas está ligado à aprendizagem. O próprio modelo de formação continuada PDE (Programa de Desenvolvimento Educacional), faz previsão sobre o uso de ambientes virtuais. Segundo Chiofi e Oliveira (2014), os ambientes virtuais são necessários ao desenvolvimento das atividades colaborativas. Como referência de bons resultados da aplicação de ambientes virtuais na educação básica pode-se citar algumas escolas do estado do Paraná. A inclusão digital nas escolas da rede pública do Paraná tem mostrado uma nova filosofia educacional por meio da ampliação da rede de inovações tecnológicas. Essa inserção proporcionou novas formas de trabalhar os conteúdos curriculares e percebeu-se uma melhor interação entre os próprios alunos e os alunos com os professores (CHIOFI; OLIVEIRA, 2014).

Com o intuito de concretizar a proposta da utilização de ferramentas computacionais na educação básica, nesta Seção é apresentada uma plataforma. Para tanto, foi criado um *website* onde são apresentadas as ferramentas da Tabela 1. Mais do que centralizar as ferramentas, a plataforma fornece informações sobre como podem ser utilizadas. Ela foi chamada de Apcode (*website*: www.apcode.com.br), este nome faz relação entre as palavras aprender e código. Na Figura 15 é apresentada a página inicial da plataforma.

Na Figura 15 é mostrado que a plataforma apresenta o conteúdo de forma clara, aumentando a capacidade de interação dos alunos. Ela possui as seguintes características: é um ambiente de aprendizagem digital; de fácil acesso ao *website*; e com navegação intuitiva. Na Figura 16 é apresentado de forma adaptada a Tabela 1 deste trabalho. Com essa centralização de informações os alunos podem escolher entre as várias ferramentas sem precisar fazer buscas avançadas ou digitar URL's complexas.

Cabe aos professores fazerem a escolha da ferramenta que deseja utilizar, de acordo com as necessidades que devem ser abordadas na aula. Outro objetivo da Apcode é orientar os alunos por quais ferramentas devem começar a aprender. Com isso, a plataforma permite uma aprendizagem progressiva e constante. Na Figura 17 é apresentada a tela onde os alunos são alertados sobre os níveis de dificuldade. Na Figura 18 é mostrada a área reservada aos professores, com informações sobre a importância do ensino de computação utilizando às ferramentas lúdicas. Na Figura 19, página "sobre", é listado os principais objetivos da plataforma. Por fim, na Figura 20 é mostrada a página "blog", onde serão disponibilizadas várias notícias relacionadas a plataforma.

Figura 15 – Plataforma Apcode: Tela inicial

APCODE
Plataforma de suporte às ferramentas de ensino de Lógica de Programação

Home Sobre ▾ Ferramentas de Ensino Blog

Uma forma divertida de aprender programação. Professores, ensinem seus alunos!

Vamos começar?

[Sou aluno](#) [Sou professor](#)

Vantagens de aprender Lógica de Programação:

- 🛠 Desenvolve a criatividade e o raciocínio lógico;
- 🎧 É uma forma divertida de aprender outras disciplinas;
- ✓ Melhora a capacidade de resolução de problemas;
- ✚ Aprender com jogos dinâmicos e trabalho em grupo. E muito mais!

Depoimentos

Professora Floresa

A programação estimula a criatividade, inspiração e inovação e tudo isso é essencial em qualquer carreira.

Chris Bosh

Se você programa, você pode escolher o rumo da sua vida.

Linda Llukas

Código de computador é a próxima linguagem universal e sua sintaxe será limitada apenas pela imaginação das próxima geração de programadores.

Professor Geraldo

A ciência da computação não é para outra pessoa. É para você, aqui e agora.

Alguns incentivadores do ensino de lógica de programação:


MIT, Google, Facebook, Twitter, BBC, Amazon Web Services, Microsoft, Bill Gates, Reid Hoffman, Mark Zuckerberg e Priscilla Chan...

Desenvolvido como projeto de dissertação do Mestrado em Educação da UFVJM - Versão Beta.
Conteúdo de terceiros devidamente citados. Todas as imagens aqui usadas têm direito de reutilização não comercial.

Fonte: Próprio autor.


Figura 16 – Plataforma Apcode: Tela onde são listadas as ferramentas de ensino

Home
Sobre
Ferramentas de Ensino
Blog


all can code

- **Desenvolvedor:** Allcancode Inc;
- **Grau de dificuldade:** Muito baixo;
- **Características:** Jogos para crianças com animações interativas.

[+ Acessar site](#)


MIT App Inventor

- **Desenvolvedor:** Google e MIT;
- **Grau de dificuldade:** Médio;
- **Características:** Ferramenta voltada para o desenvolvimento de aplicativos Android.

[+ Acessar site](#)


Blockly Games

CodeBug

Disney Jogos

GAMEBLOX

GAMEFROOT

LearnToMod

micro:bit

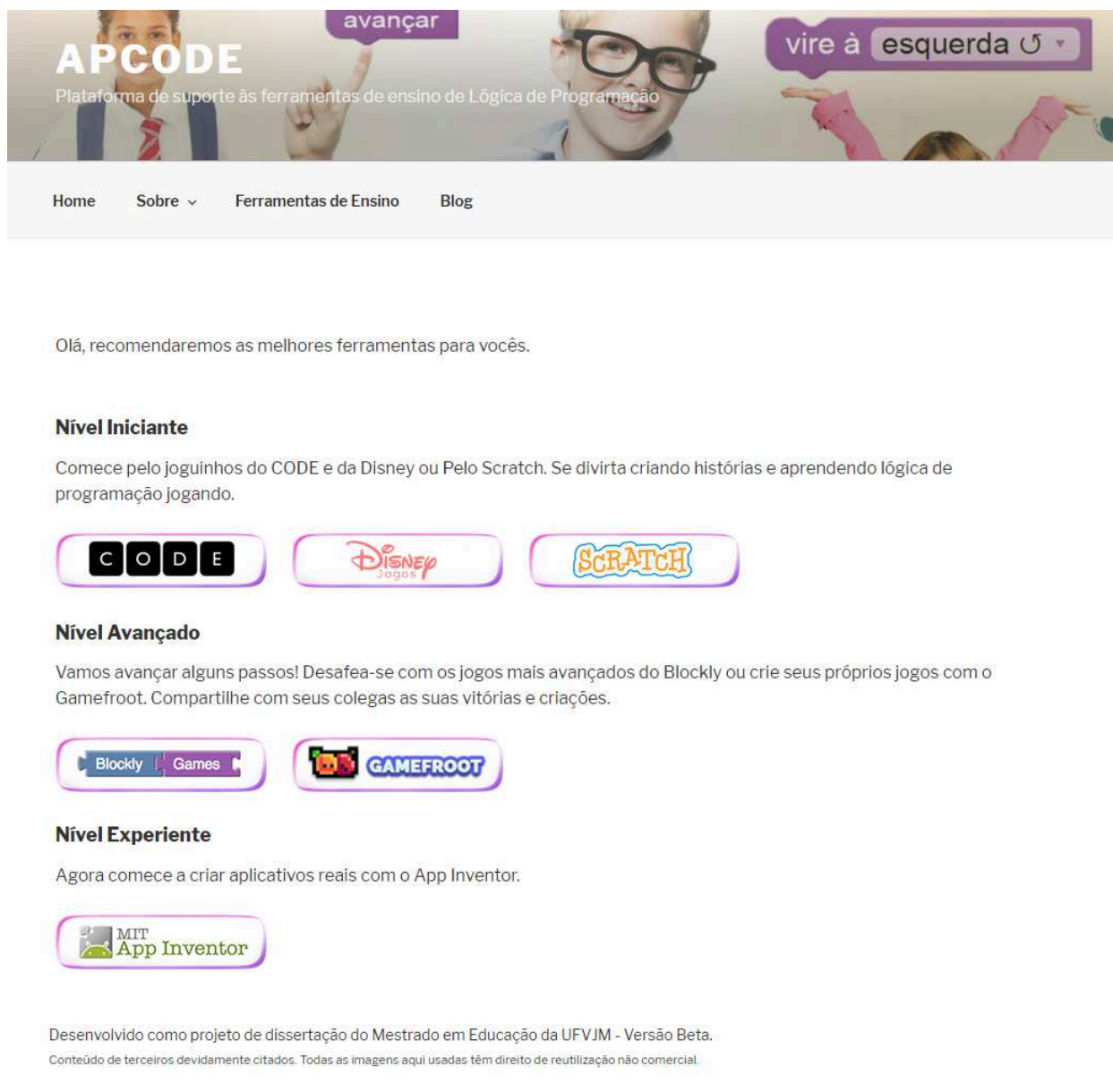
OP EN Roberta®

OzoBlockly

SCRATCH

Fonte: Próprio autor.

Figura 17 – Plataforma Apcode: Tela de suporte ao aluno



Fonte: Próprio autor.

Figura 18 – Plataforma Apcode: Tela da área reservada aos professores



Aos Professores



A Sociedade Brasileira de Computação (SBC) é uma das grandes incentivadoras da proposta de ensino de lógica de programação na educação básica. Ela recomenda que o ensino seja a partir do ensino básico para que os princípios e habilidades ensinadas contribuam com a resolução de problemas. O ensino de lógica de programação na educação básica pode ajudar os alunos a se orientarem sobre carreiras profissionais. No trabalho realizado por Scaico et al. (2012) foi utilizado o Scratch para demonstrar que os alunos utilizaram os conceitos aprendidos para entender o que acontece em um curso superior da área de computação.

É sugerido que o desenvolvimento de atividades para ensinar lógica de programação em sala de aula siga os seguintes princípios:

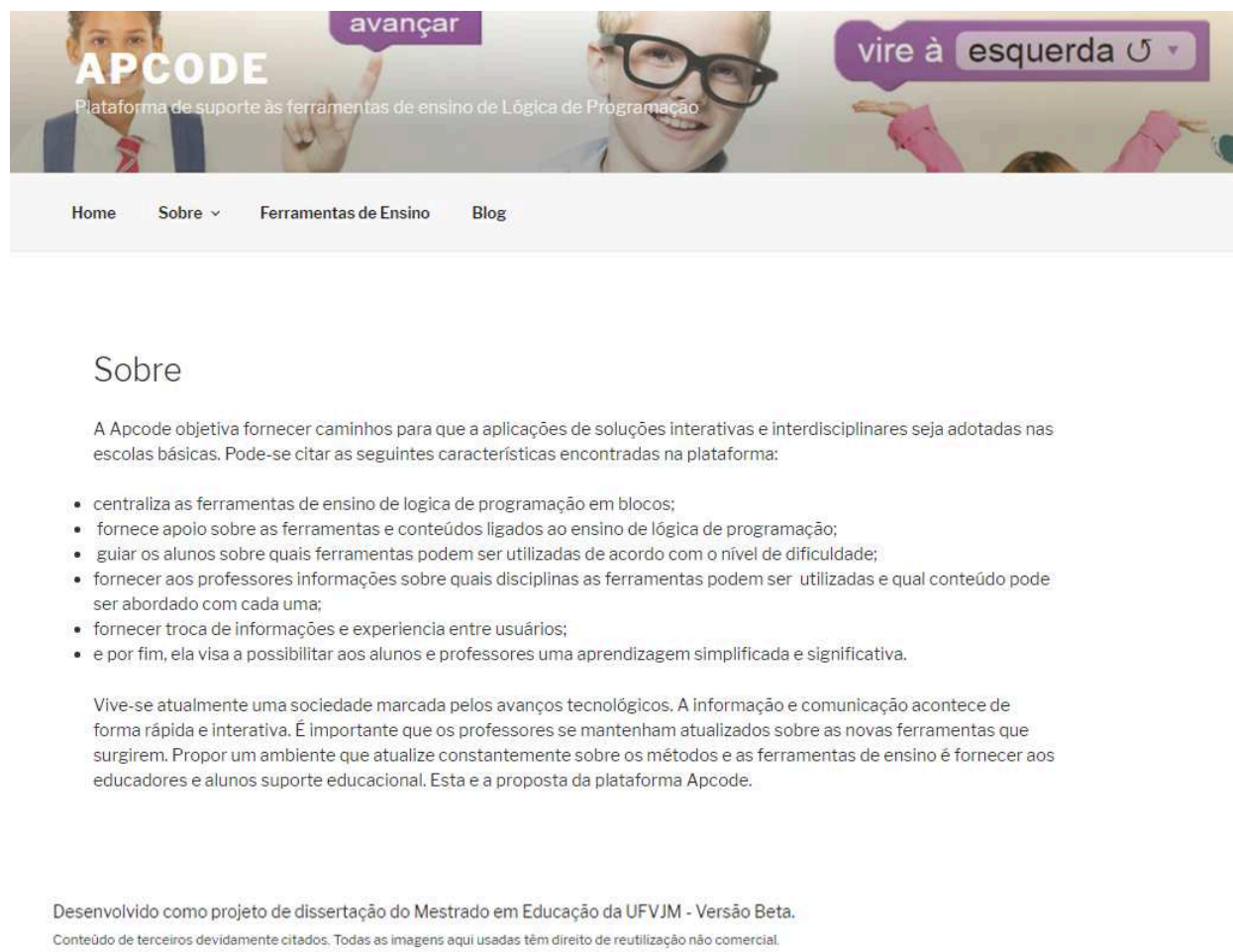
- **Ensinar um conteúdo:** é importante que ao mesmo tempo que o professor trabalhe o raciocínio lógico do aluno um conteúdo curricular seja repassado ao mesmo;
- **Manter o aluno motivado:** se a atividade não for levada até um determinado propósito o aluno pode se sentir frustrado. Toda atividade deve ter um início, meio e fim e a motivação é importante para que isso ocorra;
- **O ensino de lógica de programação deve provocar diversão:** a proposta de ensinar programação usando blocos é promover o aprendizado de forma prazerosa para o aluno.

Desenvolvido como projeto de dissertação do Mestrado em Educação da UFVJM - Versão Beta.

Conteúdo de terceiros devidamente citados. Todas as imagens aqui usadas têm direito de reutilização não comercial.

Fonte: Próprio autor.

Figura 19 – Plataforma Apcode: Tela da página sobre



Fonte: Próprio autor.

Figura 20 – Blog da plataforma Apcode



Uma maneira prazerosa de ensinar às crianças sobre computadores



"Código de computador é a próxima linguagem universal e sua sintaxe será limitada apenas pela imaginação da próxima geração de programadores. Linda Liukas está ajudando a educar crianças resolvidoras de problemas, encorajando-nas a verem computadores não como mecanismos, entediante e complicados, mas máquinas coloridas, expressivas e feitas para se modificar. Nessa palestra, ela nos convida a imaginar um mundo onde as Ada Lovelaces de amanhã cresçam otimistas e bravas frente à tecnologia e usem-na para criar um novo mundo que seja maravilhoso, encantador e um pouquinho estranho."

Fonte: TED. Acesso em: https://www.ted.com/talks/linda_liukas_a_delightful_way_to_teach_kids_about_computers?language=pt-br. Acessado em 19 de Fevereiro de 2017.

Desenvolvido como projeto de dissertação do Mestrado em Educação da UFVJM - Versão Beta.

Conteúdo de terceiros devidamente citados. Todas as imagens aqui usadas têm direito de reutilização não comercial.

Fonte: Próprio autor.

Pode-se considerar uma iniciativa inovadora, sendo que os *sites* das ferramentas geralmente só oferecem acesso à apenas uma. Quando há várias ferramentas expostas em um único local, caso do CODE, são de uma mesma equipe com características semelhantes. Isso acarreta uma uniformidade das ferramentas dispostas nos seus devidos *sites*.

A Apcode objetiva fornecer caminhos para que a aplicação de soluções interativas e interdisciplinares sejam adotadas nas escolas básicas. Pode-se citar as seguintes características encontradas na plataforma:

- centraliza as ferramentas de ensino de lógica de programação em blocos;
- fornece apoio sobre as ferramentas e conteúdos ligados ao ensino de lógica de programação;
- guia os alunos sobre quais ferramentas podem ser utilizadas de acordo com o nível de dificuldade;
- fornece aos professores informações sobre quais disciplinas as ferramentas podem ser utilizadas e qual conteúdo pode ser abordado com cada uma;
- fornece troca de informações e experiência entre usuários;
- e por fim, ela visa a possibilitar aos alunos e professores uma aprendizagem simplificada e significativa.

Vive-se atualmente em uma sociedade marcada pelos avanços tecnológicos. A informação e comunicação acontece de forma rápida e interativa. É importante que os professores se mantenham atualizados sobre as novas ferramentas que surgirem. Propor um ambiente que atualize constantemente sobre os métodos e as ferramentas de ensino é fornecer aos educadores e alunos suporte educacional. Esta é a proposta da plataforma Apcode.

8 CONSIDERAÇÕES FINAIS E TRABALHO FUTURO

Criar soluções para melhorar o ensino, seja qual for a área, é algo indispensável atualmente. Para tanto, novas metodologias de ensino estão sendo adotadas nas salas de aula. A computação deveria ser inserida no contexto escolar através de uma aprendizagem lúdica. O ensino de computação é pouco aplicado nas escolas do Brasil, na maioria das vezes é dissociado dos conceitos primordiais, dando lugar ao ensino de informática e de ferramentas básicas. Ensinar informática traz poucos benefícios ao desenvolvimento do raciocínio lógico dos alunos. Por outro lado, o ensino de computação, principalmente da lógica de programação, resultaria em melhores resultados. As ferramentas de ensino baseadas em blocos podem ser encaradas como uma boa alternativa de ensino que visam ao aprendizado dos alunos. Apresentou-se neste trabalho várias opções de ferramentas que utilizam a metodologia de blocos. Cabe ao professor avaliar quais delas são as mais adequadas para cada situação ou turma de alunos. Neste trabalho foram discutidas duas dessas ferramentas com exemplos de utilização.

Mostrou-se também que as ferramentas apresentadas podem ser utilizadas em sala de aula de forma interdisciplinar. Possibilitando ao aluno experimentar o aprendizado de forma lúdica, fazendo uso dos conceitos da Ciência da Computação. Ao mesmo tempo a interdisciplinaridade possibilita ao professor instigar a curiosidade dos alunos, despertando neles o senso crítico.

Por fim, foi desenvolvida uma plataforma (Apcode) que auxilia professores e alunos no uso das ferramentas de ensino e aprendizagem. A plataforma reúne as principais ferramentas, possibilitando ao aluno encontrar várias alternativas em um mesmo lugar e os orientando sobre quais são mais aconselháveis para cada aluno.

Como proposta de trabalho futuro pretende-se aplicar a plataforma em escolas de ensino básico, além de avaliar se haverá um avanço no desempenho dos alunos em outras disciplinas.

REFERÊNCIAS

- ABELSON, H. **App inventor for android**. USA: Google Research Blog, 2009.
- BARBIER, R. **A pesquisa-Ação. Série Pesquisa em Educação**. Brasília: Editora Plano, 2002. v. 8.
- BARCELOS, T. S.; SILVEIRA, I. F. **Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica**. Curitiba, PR: XX Workshop sobre Educação em Computação. Anais do XXXII CSBC, 2012.
- BASE NACIONAL COMUM CURRICULAR. 2016. Disponível em: <http://basenacionalcomum.mec.gov.br>. Acesso em: 27 de dezembro de 2016.
- BEGHINI, L. B. **Automação Residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android**. São Paulo, SP: [s.n.], 2013.
- BELL, T.; ANDREAE, P.; ROBINS, A. **Computer science in NZ high schools: the first year of the new standards**. Raleigh, North Carolina: [s.n.], 2012. 343–348 p.
- BLOCKLY GAMES. 2016. Disponível em: <https://blockly-games.appspot.com/about>. Acesso em: 12 de dezembro de 2016.
- CHIOFI, L. C.; OLIVEIRA, M. R. F. **O uso das tecnologias educacionais como ferramenta didática no processo de ensino e aprendizagem**. Paraná,: [s.n.], 2014.
- COMPUTING AT SCHOOL WORKING GROUP. **Computer Science: A curriculum for schools**. [S.l.], 2012.
- CORMEN, T. H. **Introduction to algorithms**. Cambridge: MIT press, 2009.
- DAVIS, M. E.; PHILLIPS, J. A. **Aprendendo PHP e MySQL**. Rio de Janeiro: A. Books, 2008. v. 12.
- DEITEL, H.; DEITEL, P. **C++ Como Programar**. 5ª. ed. Porto Alegre: Bockman, 2006.
- DEITEL, H. M.; DEITEL, P. J. **C# Como Programar**. 1ª. ed. São Paulo: Pearson Education, 2003.
- DEITEL, H. M.; DEITEL, P. J. **Java: Como Programar**. 6ª. ed. Porto Alegre: Pearson, 2005.
- DEITEL, H. M.; DEITEL, P. J.; LISTFIELD, J.; NIETO, T.; YAEGER, C.; ZLATKINA, M.; TORTELLO, J. E. N. **C: como programar**. 6ª. ed. [S.l.]: Makron Books, 2011.
- FEITOSA, T. S. **Informática na Educação: novas ferramentas pedagógicas para o professor na atualidade**. 3ª. ed. Tatuapé, São Paulo: [s.n.], 2001.
- FERREIRA, S. L. **Introduzindo a noção de interdisciplinaridade**. [S.l.]: Cortez São Paulo, 2005. v. 8. 33–35 p.
- FLANAGAN, D. **JavaScript: O Guia Definitivo”, 6a. Edição**. [S.l.]: Editora Bookman, 2012.
- FURBER, S. **Shut down or restart? The way forward for computing in UK schools**. London: [s.n.], 2012.

ISIS, P. **Escola de BH ensina robótica para crianças e adolescentes**. [s.n.], 2016. Disponível em: <http://simi.org.br/noticia/escola-de-bh-ensina-robotica-para-criancas-e-adolescentes.html>. Acesso em: 24 de dezembro de 2016.

KLEINA, N. **Escola ensina robótica e programação para crianças a partir de seis anos**. [s.n.], 2016. Disponível em: <https://pro.tecmundo.com.br/educacao/88188-escola-ensina-robotica-programacao-criancas-partir-seis-anos.htm>. Acesso em: 27 de dezembro de 2016.

LDB. 2015. Disponível em: <http://www.ufsj.edu.br/portal2-repositorio/File/proen/ldb11ed.pdf>. Acesso em: 22 de Janeiro de 2017.

MATTEI, C. **O prazer de aprender com a informática na Educação Infantil**. Blumenau, SC: [s.n.], 2011.

MELO, L. de A.; COSTA, T. K. de L.; BATISTA, A. C. D. **Pense bem: proposta e desenvolvimento de jogo digital para ensino de computação na educação básica**. [S.l.: s.n.], 2013. v. 24. 346 p.

MENEZES, N. N. C. **Introdução à programação com Python: algoritmos e lógica de programação para iniciantes**. [S.l.]: Novatec, 2010.

MIT APP INVENTOR 2. 2016. Disponível em: <http://ai2.appinventor.mit.edu>. Acesso em: 12 de Janeiro de 2016.

MORAN, J. M. **Novas tecnologias e mediação pedagógica**. 3a edição. ed. Campinas, SP: Papirus Editora, 2001.

MORIN, E. **Os setes saberes necessários à educação do futuro**. [S.l.]: Cortez Editora, 2014.

NAPOLITANO, R. L.; BATISTA, F. F. **A CIÊNCIA DA COMPUTAÇÃO APLICADA NO PERÍODO DE EDUCAÇÃO INFANTIL**. Rio de Janeiro: [s.n.], 2003.

OLSON, P. **Why Estonia Has Started Teaching Its First-Graders To Code**. 2012. [s.n.], 2012. Disponível em: <http://www.forbes.com/sites/parmyolson/2012/09/06/why-estoniahas-started-teaching-its-first-graders-to-code/>. Acesso em: 27 de dezembro de 2016.

PAPERT, S.; VALENTE, J. A.; BITELMAN, B. **Logo: computadores e educação**. [S.l.]: Brasiliense, 1980.

PIAGET, J.; BARROS, M. de. **Problemas gerais da investigação interdisciplinar e mecanismos comuns**. [S.l.: s.n.], 1973.

PIZARRO, L. **Projeto leva aulas de programação para escola pública de BH**. [s.n.], 2016. Disponível em: <http://www.otempo.com.br/interessa/tecnologia-e-games/projeto-leva-aulas-de-programa%C3%A7%C3%A3o-para-escola-p%C3%BAblica-de-bh-1.1393647>. Acesso em: 27 de dezembro de 2016.

RESNICK, M.; MALONEY, J.; MONROY-HERNÁNDEZ, A.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSENBAUM, E.; SILVER, J.; SILVERMAN, B. **Scratch: programming for all**. [S.l.]: Communications of the ACM, 2009. v. 52. 60–67 p.

ROSE, K.; ELDRIDGE, S.; CHAPIN, L. **The internet of things: An overview**. [S.l.: s.n.], 2015.

SANTOS, R. P. dos; COSTA, H. A. X. **Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática**. [S.l.: s.n.], 2006. v. 5. 41–50 p.

SCAICO, P. D.; CORLETT, E. F.; PAIVA, L. F.; RAPOSO, E. H. S.; ALENCAR, Y. **Relato da Utilização de uma Metodologia de Trabalho para o Ensino de Ciência da Computação no Ensino Médio**. Rio de Janeiro: [s.n.], 2012. v. 1.

SCAICO, P. D.; LIMA, A. A. de; AZEVEDO, S.; SILVA, J. B. B. da; RAPOSO, E. H.; ALENCAR, Y.; MENDES, J. P.; SCAICO, A. **Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch**. [S.l.: s.n.], 2013. v. 21. 92 p.

SCHLEICHER, A. **Programme for International Student Assessment (PISA): Result from PISA 2015**. [S.l.: s.n.], 2015.

SCRATCH. 2016. Disponível em: <https://scratch.mit.edu>. Acesso em: 12 de Janeiro de 2016.

TUCKER, A. **A Model Curriculum for K–12 Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee**. [S.l.]: ACM, 2003.

VALENTE, J. A. **Por que computadores na educação**. Campinas: Computadores e Conhecimento: repensando a educação, 1993. 1–23 p.

VICARI, R. M. **Ciência, Praxis e Interdisciplinaridade**. São Paulo, SP: [s.n.], 2016.

WESTPHALEN, E. F.; TONI, C. C.; SANTOS, M. A. **Informática: Um Instrumento Interdisciplinar**. Curitiba, PR: XX Congresso Nacional da Sociedade Brasileira de Computação, 2000.

WOLBER, D. **App inventor and real-world motivation**. San Francisco: Proceedings of the 42nd ACM technical symposium on Computer science education, 2011. 601–606 p.